

FOSS in Education

What education should take, and contribute to Free and Open
Source Software

Dr Colin Turner
Senior Lecturer
School of Electrical and Mechanical Engineering
c.turner@ulster.ac.uk
ct@fsfe.org

30th June 2007
BarCamp Belfast

Outline

- 1 Introduction
- 2 Consumption of FOSS
- 3 Education about FOSS
- 4 Creation of FOSS

Context of this talk

This talk is mainly focused on exploring the relationships between **FOSS**, Free and Open Source Software, and education, and more specifically, my area of knowledge : higher education.

However, many of these arguments are as relevant to the public sector in general, or indirectly publicly funded bodies.

Terms and their usage

It's worth noting at this early stage that we should not take it for granted that people understand the meaning of **free** or **open source** software. Even in the FOSS community, there are some divergences of opinion and usage here.

Pretty much everyone present at this talk understands these terms, but the slides are aimed at a wider audience, so it's worth going over this, ever so briefly...

Terms and their usage

The Free Software Foundation (<http://www.fsf.org>) defines free software as providing **you** with the following freedoms

Freedoms in Free Software

- 0 the right to run it for any purpose;
- 1 the right to study it and adapt to your needs; (which implies **open source**)
- 2 to redistribute it, so you can help others;
- 3 to release improvements, so everyone benefits.

Specifically, note, **money** is not mentioned, but freedom 2 and 3 normally imply freedom from cost.

Terms and their usage

Other types of software include

- **Open Source:**

Most people use this term synonymously with free, but some mischievous people use this to literally mean, read access to the source.

- **Custom :**

Software created “in house” which isn’t being licensed for distribution - closed source, but without ethical entanglements normally.

... and

Terms and their usage

- **Proprietary** :

The “normal” model, software that is licensed, often in such extraordinary ways that as it is observed in Pratchett and Gaiman’s *Good Omens*, hell could learn from the way in which money is exchanged for so little rights. Having said that, again money might not be involved at all here, a lot of “freeware” and “shareware” software still grant no freedoms.

Back to the main thread

There are a number of ways in which education could, and perhaps should, relate to FOSS.

Relationships of Education to FOSS

1 As a **consumer**.

Clearly, education procures vast quantities of software, should some of it be FOSS?

2 As an **educator**.

Education should cover business models, ethics and technical methodologies of FOSS.

3 As a **creator**.

In some cases education can make a positive impact in the creation of FOSS. Note that a broader definition that just “code” for software should be used here.

Why not use more FOSS?

It seems obvious that education should be in favour of FOSS, the software is usually free of charge, and you have a lot of rights over it to customize it to need, and as the software is normally based in open standards, vendor lock in is avoided. Despite this, uptake of FOSS by higher education is very limited, and this is probably even more acute in the secondary education market. So it's useful to explore the reasons why this is.

Why not use more FOSS?

Here are some viewpoints that are typical in the sector, some are more valid than others.

Perceived problems with uptake of FOSS

- **Quality** “FOSS is flakey”;
- **Support** “We need support contracts”;
- **Business** “We need to use what the real world uses”;
- **Gaps** “There is no FOSS program to do X”;
- **Special Licenses** “But we get all this at a huge discount anyway”;
- **Budgets** “If we spend less, we become less important”;
- **Transparency and Accountability** “Pardon?”.

Why not use more FOSS?

Quality

FOSS products are seen as lacking in QA measures, flakey and unreliable. Of course, this is as true in the proprietary sphere, but there you simply can't do anything if the company goes out of business, loses interest etc.. It's useful to deal with case by case, much of the infrastructure of a University's IT system will probably run on, rock solid reliable, FOSS. Many GNU/Linux distributions as well as other major FOSS projects like Mozilla have elaborate QA and excellent stability.

Why not use more FOSS?

Support

Perhaps the greatest fallacy. Support networks for FOSS products are often extremely rich, and superior to those of proprietary products. For example compare Debian's "for free" support, with similar features available for MS Windows. Also support **can** be paid for, from any number of different sources, unlike in the proprietary model where support is often only available from the vendor.

Why not use more FOSS?

Business

There is a need to prepare students for “real world” software. Personally, I think Universities should provide a slight counter pressure against naked commercial pressures. However, a stronger argument is that we are supposed to be teaching transferable skills, cf OpenOffice and MS Office.

Gaps

This is a serious problem, there are indeed acute gaps in specialist areas, like for example high quality CAD. The knock on effect for a few hundred students out of thousands should not be underestimated. It means, that for a consistent desktop (which institutions try hard to obtain for maintainability) you end up with MS Windows. More follows from that

Why not use more FOSS?

Special Licenses

It's relatively cheap to obtain education licenses. However, it's not all about money, but in any case we know **why** it's cheap. To allow companies to protect market dominance and also since they know when the students graduate obtaining the software we have trained them for will be anything but cheap.

Budgets

I doubt anyone will claim this one, but at a high level, there might be some who purchase software at a corporate level because the budget gives their department more power. If this issue exists, it is hard to counter, since it won't be publicly exposed.

Why not use more FOSS?

Transparency and Accountability

This seems to go over the heads of many, but a basic issue is having software that implements closed standards with public data. Who owns the data? How will you access it in 20 years time? What is the software actually doing with it?

When you commission software do you own any rights? The copyright to the code? If not, does the license mean only the original authors can change it for you?

These are concerns many simply don't have, **but they should!**

What education should be saying about FOSS

What should we be teaching?

- **Business Models**; everyone affected should understand both proprietary and FOSS software models.
- **Ethics and Social Models**; FOSS has interesting ethical origins in many cases and is also rich in social dynamics.
- **Distributed project management, QA**; projects managed over an entire planet have a rather different perspective on these issues.
- **Technical Methodologies**; following from the above, there are certain technologies that make this all possible.

How can we create more FOSS?

What should we be creating?

- **Final Year Projects;**
Every year, hundreds of software students are looking for worthy projects for final year projects, masters and even doctoral projects. These could fill the “gaps” mentioned above.
- **Custom needs;**
Universities create custom applications for their own needs. Why not make them FOSS?
- **Procurement;**
Before buying expensive packages, can we write them?
Can we augment existing software?

Capstone Projects

We can identify a number of gaps in the FOSS market. For example, in the field of CAD. As previously explored, such gaps have a disproportionate effect, pushing consistent platform delivery away from FOSS.

Consider that, over years, project students might contribute to such a project. Each student might be given one aspect of work, data storage, rendering etc., or multiple students might work on the same area to produce novel solutions. Eventually you might be in a position to close the gap.

Alternatively, look at existing FOSS projects in which a student might be interested to create a broader contribution. There are real benefits to the student:

Capstone Projects

Benefits to student working on FOSS

- Understanding of social and ethical considerations;
- Real world problems provide opportunities for deep learning and real achievement;
- Deeper understanding of key technologies and methodologies;
- Opportunities to network with “real” employers and employees.

Custom Projects

Universities frequently have a need to produce software in-house to undertake certain tasks.

This would generally be custom, non distributed, software, so there's little harm in that. What are the benefits of making such work FOSS?

Custom Projects

Benefits to making custom software FOSS

- **Gift Culture**; giving software away and making it publicly available normally acts as a powerful incentive to improve quality;
- **Open in a box**; even the impact of visibility of a project within its parent institution cannot be overlooked. Other developers can provide good ideas, or learn them. A “joined up” philosophy is more likely.
- **Prestige**; if you are going to develop the software, and not exploit it with a proprietary license, then why not show off your good practice;

... continued

Custom Projects

Benefits to making custom software FOSS

- **Dissemination**; thousands of lines of code can be more valuable than journal articles;
- **Low risk business model**; support can be provided on a consultancy basis, with no obligation.
- **Gain expertise from others**; encourage other institutions to improve your software at little or no man power costs to you;
- **Distribute expertise**; if your core team leave your control, you still have expertise in the code existing elsewhere (sustainability).

In short, FOSS, is an excellent exploitation model.

What have we done?

So, what are we doing at the University of Ulster?

Activity at UU

- We have made free (under the GPL) two of our large institutional IT systems that are nationally recognised for good practice. We host that at <http://foss.ulster.ac.uk>.
- We have talked across the UK to publicise this.
- We have hosted a conference inviting both practitioners and developers, from both internal guests, and over 30 external institutions across the UK.
- We have people requesting the software and consultancy deals for support.

What have we done?

Activity at UU

- We are seeing a few people interested in the development effort.
- We are starting to bid for funding for more FOSS development.
- We are talking to industry about Open Source matters.
<http://foss.ulster.ac.uk/mediawiki/index.php/Ossni>
- We have set up a Web Developers Interest Group, which has seen a number of talks about work with FOSS technologies inside and outside the University.

What have we done?

This will hopefully catalyse more change. Now the University has tried the model and found it successful, it is increasingly likely to support it in the future.

We plan to start a “local chapter” of the Belfast Linux Users Group in the University too.

Summary

- 1 Consumption of FOSS
 - FOSS procurement in the education sector is limited by many misconceptions.
 - Most of these are easily debunked.
- 2 Education about FOSS
 - FOSS technology and philosophy need to be taught.
 - This is an improving picture, but more needs done here.
- 3 Creation of FOSS
 - Universities can be a hotbed for FOSS creation.
 - There are reasons for optimism that this is increasingly being recognised, and universities are understanding the benefit to them.

Summary

So what should we do?

- 1 Keep explaining what FOSS is to everyone.
- 2 Explain why they should care.
- 3 Educate our students about the issues.
- 4 Encourage them to take part.
- 5 Persuade management why it benefits them.
- 6 Reach out to other institutions.
- 7 Apply for grants. e.g. JISC.
- 8 Secondary education is very stitched up. Consider helping at primary and higher levels first.